View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Online Research Database In Technology

DTU Library

Numerical Modelling of Heat Transfer using the 3D-ADI-DG Method - with Application for Pultrusion.

Technical report

Rasmussen, Filip Salling; Klingaa, Christopher Gottlieb; Sonne, Mads Rostgaard; Hattel, Jesper Henri

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Rasmussen, F. S., Klingaa, C. G., Sonne, M. R., & Hattel, J. H. (2019). Numerical Modelling of Heat Transfer using the 3D-ADI-DG Method - with Application for Pultrusion.: Technical report. Kgs. Lyngby: Technical University of Denmark (DTU).

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Technical University of Denmark

Department of Mechanical Engineering

Numerical Modelling of Heat Transfer using the 3D-ADI-DG Method

- with Application for Pultrusion.

Technical Report

by

Filip Salling Rasmussen

Co-authors: Christopher G. Klingaa,
Mads R. Sonne and Jesper H. Hattel
Section for Manufacturing Engineering

1st Edition, March 2019

Copyright ©, Filip Salling Rasmussen, 2019
Department of Mechanical Engineering
Technical University of Denmark
DK 2800 Kgs. Lyngby

1 Introduction

The objective of the present report is to give a thorough description on how to implement a 3-Dimensional heat transfer solver applicable for the pultrusion process. The solver will use the Alternate Direction Implicit method modified by Douglass & Gunn (3D-ADI-DG) [1]. A similar solver was introduced for the first time in this specific field of application by Baran et al. [2]. The current report does however serve as a thorough introduction for the reader to create their own solver with the associated benefits, e.g. understanding and control of all aspects of the solver. The solver was developed, verified and validated following previous work by the authors [4] and used to investigate the effect of varying fiber volume fraction distribution on the cure behaviour during the pultrusion process [5].

The Pultrusion Process

Pultrusion is a continuous process for manufacturing of Fiber Reinforced Polymers (FRP) with constant cross-section. The fibers are either impregnated in a resin bath or by resin injection directly in the die - the latter is illustrated in Fig. 1.1. The impregnated fibers are drawn through the heating/forming die where heaters ensure proper curing of the thermoset resin.

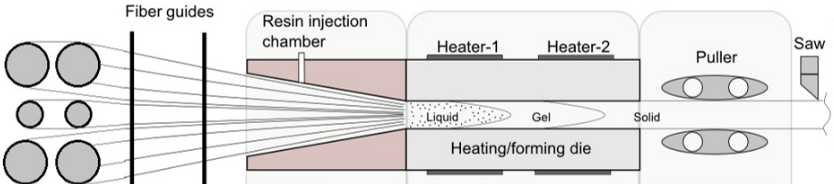


Figure 1.1: 2D illustration of the resin injection pultrusion process [4].

It will be assumed that the fibers are fully impregnated and no significant curing have happened before the heating/forming die. Hence, the numerical solution is the discretized temperature field in the heating/forming die and the FRP composite. The reader is made aware that the present report will solely focus on heat transfer modelling and the reader is referred to [2] and [4] for a description on how to include the chemical modelling, i.e. curing.

2 Heat Transfer Modelling - Pultrusion

The governing equation for the convection-diffusion problem at hand is the energy equation

$$\frac{\partial \left(\rho c_p T\right)}{\partial t} + \nabla \cdot \left(\rho c_p T \mathbf{u}\right) - \nabla \cdot \left(\lambda \nabla T\right) = S_T^{'''} \tag{2.1}$$

where ρ is density in $\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$, c_p is specific heat capacity in $\left[\frac{\mathrm{J}}{\mathrm{kgK}}\right]$, T is temperature in [°C], t is time in [s], $\mathbf{u}$ is velocity in $\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$, λ is the thermal conductivity (tensor) in $\left[\frac{\mathrm{W}}{\mathrm{mK}}\right]$ and $S_T^{'''}$ is a volumetric source term in $\left[\frac{\mathrm{W}}{\mathrm{m}^3}\right]$. The effective material properties are calculated using the rule of mixtures and assumed constant. The following notation is equivalent to (2.1) and will be used to ease the understanding of the time discretization in section 2.2.

$$\dot{Q} + \mathbf{q}_u - \mathbf{q}_\lambda = S_T^{'''} \tag{2.2}$$

In (2.2): $\dot{Q}$ is the change of heat content over time and $\mathbf{q}_u$, $\mathbf{q}_\lambda$ denotes the convective and diffusive heat flux, respectively. In the current work, energy conservation, as described in (2.1), will be modelled using a control volume based finite difference model (CV-FDM). The spatial discretization, the choice of time scheme, the implementation algorithm and an example of simulation results will be presented in the following four sections.

2.1 Spatial Discretization

The principal of energy conservation is used to set up the discretized version of the energy equation. Fig. 2.1 illustrates the 3D notation used henceforth. The material properties ρ, c_p and λ will be assumed constant. The discretization will account for a non-uniform but structured mesh and will only consider convection in the pulling direction (x) and further assume this velocity to be positive and constant. Hence, $\mathbf{u} = [u_x, 0, 0] = const.$ and $u_x > 0$. The diffusive heat fluxes will be evaluated using a 3-point Central Difference Scheme (CDS), while the convective heat flux will be evaluated using a 2 point Upwind Difference Scheme (UDS). Hence, the discretized energy equation over a control volume $\Delta x \Delta y \Delta z$ becomes

$$\begin{gathered}
\rho c_p \frac{\partial T}{\partial t} \Delta x \Delta y \Delta z \\
+ \rho c_p u_x \frac{T_P - T_W}{\Delta x_w} \Delta x \Delta y \Delta z \\
- \begin{Bmatrix}
\lambda_x \frac{\Delta y \Delta z}{\Delta x_e} (T_E - T_P) - \lambda_x \frac{\Delta y \Delta z}{\Delta x_w} (T_P - T_W) + \\
\lambda_y \frac{\Delta x \Delta z}{\Delta y_b} (T_B - T_P) - \lambda_y \frac{\Delta x \Delta z}{\Delta y_f} (T_P - T_F) + \\
\lambda_z \frac{\Delta x \Delta y}{\Delta z_n} (T_N - T_P) - \lambda_z \frac{\Delta x \Delta y}{\Delta z_s} (T_P - T_S)
\end{Bmatrix} \\
= S_T^{'''} \Delta x \Delta y \Delta z
\end{gathered} \tag{2.3}$$

The reader is encouraged to realise that (2.3) is equivalent to (2.1) - let (2.3) be divided with the volume and subsequently let Δx, Δy and Δz approach zero.

The choice of time stepping scheme will be explained in detail in the following subsection. Hence, for now the time dependent term in (2.3), related to the change of heat content, will simply be discretized as $\Delta x \Delta y \Delta z \cdot \rho c_p \cdot \frac{\Delta T_P}{\Delta t}$.

For the purpose of numerical implementation, proper weight functions are introduced and (2.3) can be written as

$$\begin{gathered}
H_0 \Delta T_P \\
+ H_u (T_P - T_W) \\
- \begin{Bmatrix}
H_e (T_E - T_P) - H_w (T_P - T_W) + \\
H_b (T_B - T_P) - H_f (T_P - T_F) + \\
H_n (T_N - T_P) - H_s (T_P - T_S)
\end{Bmatrix} \\
= S_T
\end{gathered} \tag{2.4}$$

where $S_T = S_T^{'''} \Delta x \Delta y \Delta z$, and the weight functions are defined in the following way.

The capacity weight function:

$$H_0 = \frac{\rho c_p \cdot \Delta x \Delta y \Delta z}{\Delta t} \tag{2.5}$$

The convective weight function:

$$H_u = \rho c_p u_x \Delta y \Delta z \tag{2.6}$$

The conductivity weight functions in the x, y and z direction, respectively, using the notation illustrated in fig. 2.1 and 2.2:

$$\begin{aligned} H_w &= \lambda_x \tfrac{\Delta y \Delta z}{\Delta x_w} \\ H_e &= \lambda_x \tfrac{\Delta y \Delta z}{\Delta x_e} \end{aligned} \tag{2.7}$$

$$\begin{aligned} H_f &= \lambda_y \tfrac{\Delta x \Delta z}{\Delta y_f} \\ H_b &= \lambda_y \tfrac{\Delta x \Delta z}{\Delta y_b} \end{aligned} \tag{2.8}$$

$$\begin{aligned} H_s &= \lambda_z \tfrac{\Delta x \Delta y}{\Delta z_s} \\ H_n &= \lambda_z \tfrac{\Delta x \Delta y}{\Delta z_n} \end{aligned} \tag{2.9}$$

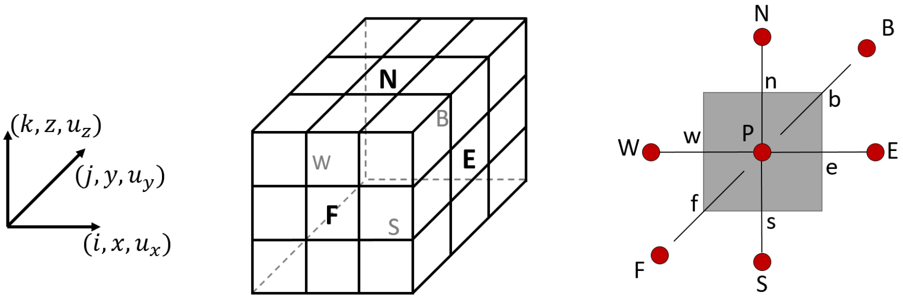
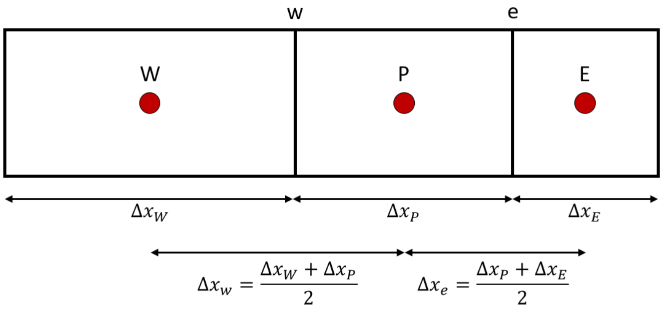


Figure 2.1: Schematic illustration of the 3D finite volume grid with labelling of the CV neighbours and CV faces.



w
e
W
P
E
Δx_W
Δx_P
Δx_E
$\Delta x_w = \frac{\Delta x_W + \Delta x_P}{2}$
$\Delta x_e = \frac{\Delta x_P + \Delta x_E}{2}$

Figure 2.2: Schematic illustration of notation used for the structured non uniform mesh (x-direction).

2.2 Time Discretization

The numerical discretization in time will be conducted using the Alternate Direction Implicit method modified by Douglas & Gunn (ADI-DG). The original 3D-ADI method implies dividing each time step into three minor, consecutive time sub-steps equal in size [3]. In the modification by Douglas & Gunn the three time sub-steps are not equal in size and should hence, to some extend, be interpreted as overlapping - this is illustrated in fig. 2.3. The advantage of using the modified 3D-ADI-DG method is good stability even for problems with a large convection to diffusion heat transfer ratio (i.e. large Peclet number, $Pe > 2$, see [2]) - which is typically the case for pultrusion of FRPs [4].

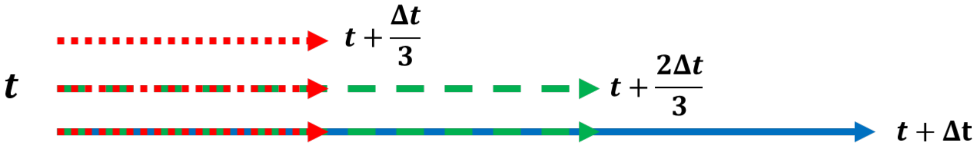


Figure 2.3: Illustration of the time sub stepping scheme for the ADI-DG algorithm.

For simplicity the direction chosen to be implicit for each time sub-step are x, y and z in chronological order. Hence, for each of the three time sub-steps the energy equation (2.2) could be written as follows.

Sub-step I:

$$\dot{Q}^{t\rightarrow t+\frac{\Delta t}{3}} + q_u^{t\rightarrow t+\frac{\Delta t}{3}} - \left(q_x^{t\rightarrow t+\frac{\Delta t}{3}} + q_y^t + q_z^t\right) = S_T^t \tag{2.10}$$

Sub-step II:

$$\dot{Q}^{t\rightarrow t+\frac{2\Delta t}{3}} + q_u^{t\rightarrow t+\frac{\Delta t}{3}} - \left(q_x^{t\rightarrow t+\frac{\Delta t}{3}} + q_y^{t\rightarrow t+\frac{2\Delta t}{3}} + q_z^t\right) = S_T^t \tag{2.11}$$

Sub-step III:

$$\dot{Q}^{t\rightarrow t+\Delta t} + q_u^{t\rightarrow t+\frac{\Delta t}{3}} - \left(q_x^{t\rightarrow t+\frac{\Delta t}{3}} + q_y^{t\rightarrow t+\frac{2\Delta t}{3}} + q_z^{t\rightarrow t+\Delta t}\right) = S_T^t \tag{2.12}$$

It is important to notice the unequal size of the three sub-steps, which cause the capacity weight function (2.5) to change for each sub-step. Hence, introducing the following three sub-step capacity weight functions:

$$H_{0x} = \frac{\rho c_p \cdot \Delta x \Delta y \Delta z}{\Delta t/3} \tag{2.13}$$

$$H_{0y} = \frac{\rho c_p \cdot \Delta x \Delta y \Delta z}{2\Delta t/3} \tag{2.14}$$

$$H_{0z} = \frac{\rho c_p \cdot \Delta x \Delta y \Delta z}{\Delta t} \tag{2.15}$$

The notation for the sub-step capacity weight functions is summarized in Table 1 together with a short notation for the sub-steps - this notation will be used henceforth.

	old	sub step I	sub step II	sub step III
Time	t	$t + \frac{\Delta t}{3}$	$t + \frac{2\Delta t}{3}$	$t + \Delta t$
Notation	t	t_x	t_y	t_z
Capacity weight functions		H_{0x}	H_{0y}	H_{0z}

Table 1: Notation used for the ADI-DG time scheme and the sub-step capacity weight functions.

2.3 Implementation of 3D-ADI-DG

The ADI-DG method involves setting up a tri-diagonal matrix system for every row or column of control volumes (CV) in each of the three major directions. Hence, the following matrix system is obtained

$$\mathbf{A} \cdot T = b \tag{2.16}$$

where $\mathbf{A}$ is the tri-diagonal coefficient matrix, T is a vector with the unknown temperatures of the row/column of CVs and b is the the right hand side (RHS) vector. The coefficients in the matrix $\mathbf{A}$ and the known values in the RHS vector b is obtained from (2.4) with the time stepping schemes illustrated in Fig. 2.3 and written in (2.10)-(2.12). The coefficients in the first and last row of the system matrix $\mathbf{A}$ as well as the first and last element in b will be defined by either a Dirichlet or a Neumann boundary condition as described in [3]. The final matrix system (2.16) is then readily solved using e.g. the *backslash* operator in MATLAB[1].

In the following let n_x, n_y, n_z and i, j, k represent the number of CVs and the index in the three spatial directions x, y, z, respectively.

Sub-step I: Implicit formulation in the x-direction.

This sub-step goes form t to $t + \frac{\Delta t}{3}$; see (2.10), Fig. 2.3 and Table 1. For each combination of $j = [2 : n_y - 1]$ and $k = [2 : n_z - 1]$, running through all inner CVs in the x-direction, $i = [2 : n_x - 1]$, a coefficient matrix $\mathbf{A}_x$ of size $n_x \times n_x$ and a RHS vector b_x with n_x elements are set up by evaluating (2.4) using the time scheme (2.10). Hence,

$$\begin{gathered}
H_{0x}\left(T_P^{t_x} - T_P^t\right) \\
+\tfrac{1}{2}H_u\left\{T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t\right\} \\
+\tfrac{1}{2}H_w\left\{T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t\right\} + \tfrac{1}{2}H_e\left\{T_P^{t_x} - T_E^{t_x} + T_P^t - T_E^t\right\} \\
+H_f\left(T_P^t - T_F^t\right) + H_b\left(T_P^t - T_B^t\right) \\
+H_s\left(T_P^t - T_S^t\right) + H_n\left(T_P^t - T_N^t\right) \\
= S_T^t
\end{gathered} \tag{2.17}$$

where the lower-diagonal (LD), diagonal (D) and upper-diagonal (UD) elements of the coefficient matrix A_x are directly given by the coefficients of $T_W^{t_x}$, $T_P^{t_x}$ and $T_E^{t_x}$, respectively. The RHS vector b_x are all the explicit parts of (2.17)[2]. Hence, the tri-diagonal coefficients in A_x and the RHS vector elements in b_x are given by

[1] https://se.mathworks.com/help/matlab/ref/mldivide.html

[2] parts evaluated at t

$$A_x^{LD} = -\frac{1}{2}H_u - \frac{1}{2}H_w \tag{2.18}$$

$$A_x^{D} = H_{0x} + \frac{1}{2}H_u + \frac{1}{2}H_w + \frac{1}{2}H_e \tag{2.19}$$

$$A_x^{UD} = -\frac{1}{2}H_e \tag{2.20}$$

$$b_x = \left\{ \begin{array}{c} H_{0x}T_P^t \\ -\frac{1}{2}H_u \left\{T_P^t - T_W^t\right\} \\ +\frac{1}{2}H_w \left\{T_W^t - T_P^t\right\} + \frac{1}{2}H_e \left\{T_E^t - T_P^t\right\} \\ +H_f \left(T_F^t - T_P^t\right) + H_b \left(T_B^t - T_P^t\right) \\ +H_s \left(T_S^t - T_P^t\right) + H_n \left(T_N^t - T_P^t\right) \\ +S_T^t \end{array} \right\} \tag{2.21}$$

Sub-step II: Implicit formulation in the y-direction.

This sub-step goes form t to $t + \frac{2\Delta t}{3}$; see (2.11), Fig. 2.3 and Table 1. For each combination of $i = [2 : n_x - 1]$ and $k = [2 : n_z - 1]$, running through all inner CVs in the y-direction, $j = [2 : n_y - 1]$, a coefficient matrix $\mathbf{A}_y$ of size $n_y \times n_y$ and a RHS vector b_y with n_y elements are set up by evaluating (2.4) using the time scheme (2.11). Hence,

$$\begin{array}{c} H_{0y}\left(T_P^{t_y} - T_P^t\right) \\ +\frac{1}{2}H_u \left\{T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t\right\} \\ +\frac{1}{2}H_w \left\{T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t\right\} + \frac{1}{2}H_e \left\{T_P^{t_x} - T_E^{t_x} + T_P^t - T_E^t\right\} \\ +\frac{1}{2}H_f \left\{T_P^{t_y} - T_F^{t_y} + T_P^t - T_F^t\right\} + \frac{1}{2}H_b \left\{T_P^{t_y} - T_B^{t_y} + T_P^t - T_B^t\right\} \\ +H_s \left(T_P^t - T_S^t\right) + H_n \left(T_P^t - T_N^t\right) \\ = S_T^t \end{array} \tag{2.22}$$

where the lower-diagonal (LD), diagonal (D) and upper-diagonal (UD) elements of the coefficient matrix A_y are directly given by the coefficients of $T_F^{t_y}$, $T_P^{t_y}$ and $T_B^{t_y}$, respectively. The RHS vector b_y are all the explicit parts of (2.22)[3]. Hence, the tri-diagonal coefficients in A_y and the RHS vector elements in b_y are given by

$$A_y^{LD} = -\frac{1}{2}H_f \tag{2.23}$$

$$A_y^{D} = H_{0y} + \frac{1}{2}H_f + \frac{1}{2}H_b \tag{2.24}$$

$$A_y^{UD} = -\frac{1}{2}H_b \tag{2.25}$$

[3] parts evaluated at t and t_x

$$b_y = \left\{ \begin{matrix} H_{0y} T_P^t \\ -\frac{1}{2} H_u \left\{ T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t \right\} \\ +\frac{1}{2} H_w \left\{ T_W^{t_x} - T_P^{t_x} + T_W^t - T_P^t \right\} + \frac{1}{2} H_e \left\{ T_E^{t_x} - T_P^{t_x} + T_E^t - T_P^t \right\} \\ +\frac{1}{2} H_f \left\{ T_F^t - T_P^t \right\} + \frac{1}{2} H_b \left\{ T_B^t - T_P^t \right\} \\ +H_s \left(T_S^t - T_P^t \right) + H_n \left(T_N^t - T_P^t \right) \\ +S_T^t \end{matrix} \right\} \tag{2.26}$$

Sub-step III: Implicit formulation in the z-direction.

This sub-step goes form t to $t + \Delta t$; see (2.12), Fig. 2.3 and Table 1. For each combination of $i = [2 : n_x - 1]$ and $j = [2 : n_y - 1]$, running through all inner CVs in the z-direction, $k = [2 : n_z - 1]$, a coefficient matrix $\mathbf{A}_z$ of size $n_z \times n_z$ and a RHS vector b_z with n_z elements are set up by evaluating (2.4) using the time scheme (2.12). Hence,

$$\begin{matrix} H_{0z} \left(T_P^{t_z} - T_P^t \right) \\ +\frac{1}{2} H_u \left\{ T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t \right\} \\ +\frac{1}{2} H_w \left\{ T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t \right\} + \frac{1}{2} H_e \left\{ T_P^{t_x} - T_E^{t_x} + T_P^t - T_E^t \right\} \\ +\frac{1}{2} H_f \left\{ T_P^{t_y} - T_F^{t_y} + T_P^t - T_F^t \right\} + \frac{1}{2} H_b \left\{ T_P^{t_y} - T_B^{t_y} + T_P^t - T_B^t \right\} \\ +\frac{1}{2} H_s \left\{ T_P^{t_z} - T_S^{t_z} + T_P^t - T_S^t \right\} + \frac{1}{2} H_n \left\{ T_P^{t_z} - T_N^{t_z} + T_P^t - T_N^t \right\} \\ = S_T^t \end{matrix} \tag{2.27}$$

where the lower-diagonal (LD), diagonal (D) and upper-diagonal (UD) elements of the coefficient matrix A_z are directly given by the coefficients of $T_S^{t_z}$, $T_P^{t_z}$ and $T_N^{t_z}$, respectively. The RHS vector b_z are all the explicit parts of (2.27)[4]. Hence, the tri-diagonal coefficients in A_z and the RHS vector elements in b_z are given by

$$A_z^{LD} = -\frac{1}{2} H_s \tag{2.28}$$

$$A_z^{D} = H_{0z} + \frac{1}{2} H_s + \frac{1}{2} H_n \tag{2.29}$$

$$A_z^{UD} = -\frac{1}{2} H_n \tag{2.30}$$

$$b_z = \left\{ \begin{matrix} H_{0z} T_P^t \\ -\frac{1}{2} H_u \left\{ T_P^{t_x} - T_W^{t_x} + T_P^t - T_W^t \right\} \\ +\frac{1}{2} H_w \left\{ T_W^{t_x} - T_P^{t_x} + T_W^t - T_P^t \right\} + \frac{1}{2} H_e \left\{ T_E^{t_x} - T_P^{t_x} + T_E^t - T_P^t \right\} \\ +\frac{1}{2} H_f \left\{ T_F^{t_y} - T_P^{t_y} + T_F^t - T_P^t \right\} + \frac{1}{2} H_b \left\{ T_B^{t_y} - T_P^{t_y} + T_B^t - T_P^t \right\} \\ +\frac{1}{2} H_s \left\{ T_S^t - T_P^t \right\} + \frac{1}{2} H_n \left\{ T_N^t - T_P^t \right\} \\ +S_T^t \end{matrix} \right\} \tag{2.31}$$

[4] parts evaluated at t, t_x and t_y

2.4 Results

The 3D-ADI-DG model has been used for a thick glass fiber-polyurethane composite beam with a square cross-section [5]. The steady state temperature and degree of cure predictions at chosen cross-section positions are shown in Fig. 2.4. The reader should notice how the outer parts of the profile (corner and facecenter) is heated and cured predominantly in the first half of the die while curing of the inner parts (quarter diagonal and center) are taking place over the full die length.

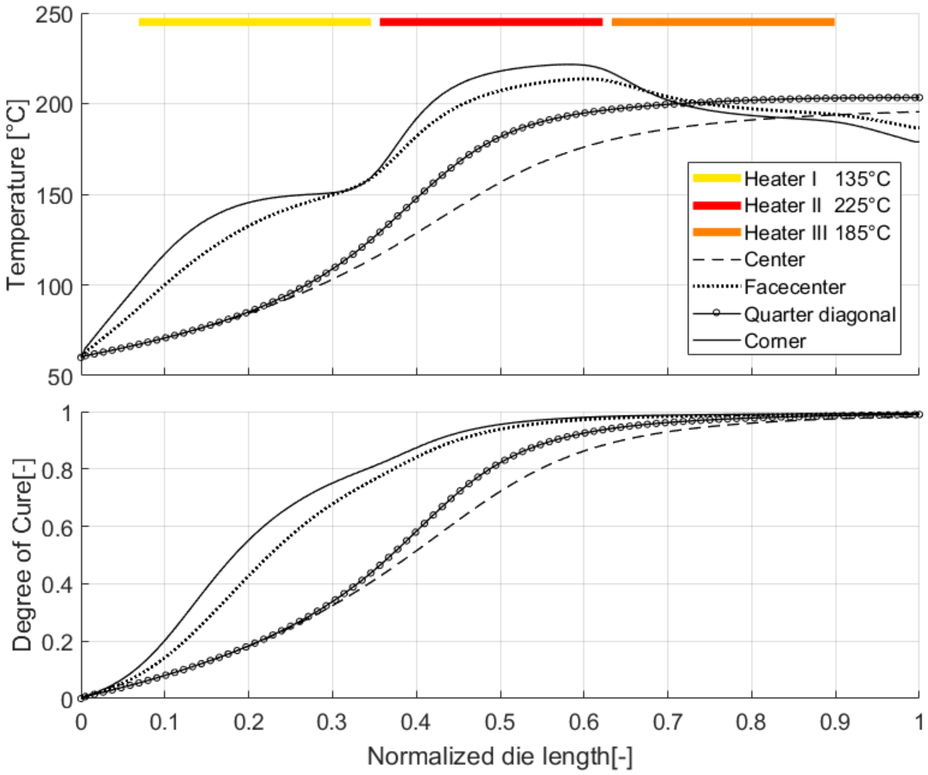


Figure 2.4: Numerical steady state temperature and degree of cure predictions inside a glass fiber-polyurethane pultruded beam during manufacturing.

3 Conclusion

The current report has presented a detailed guide on how to obtain a numerical solution to a 3-dimensional heat transfer problem using the ADI-DG method (considering convection solely in one of the major spatial directions). A 3-point central difference scheme and a 2-point upwind difference scheme were used to discretized the diffusive and the convective heat fluxes, respectively. The proposed ADI-DG solver was applied for the first time for pultrusion by Baran et al. [2] and later implemented, validated and used by the authors in [4] and [5].

Bibliography

[1] J. Douglas Jr., J.E. Gunn: A General Formulation of Alternating Direction Methods - Part I. Parabolic and Hyberbolic Problems. *Numerische Mathematik*, 6:428-453, 1964.

[2] I. Baran, J.H. Hattel, Cem C. Tutum: Thermo-Chemical Modelling Strategies for the Pultrusion Process. *Appl. Compo. Mater.*, 20:1247-1263, 2013.

[3] J.H. Hattel: *"Fundamental of Numerical Modelling of Casting Processes"*, 1. edition 2005, Polyteknisk Forlag Kgs. Lyngby, ISBN 13: 978-87-502-0969-0, ISBN 10: 87-502-0969-8.

[4] F.S. Rasmussen, M.R. Sonne, M. Larsen, J. Spangeberg, L.T. Lilleheden and J.H.Hattel: 2D Numerical Modelling of thge Resin Injection Pultrusion Process Including Experimental Resin Kinetics and Temperature Validation. *Proceeding: ICCM21*, 2017.

[5] F.S. Rasmussen, C.G. Klingaa, M. Larsen, M.R. Sonne, J. Spangenberg and J.H. Hattel: Modelling the Effect of Non-Uniform Fibre Distribution on the Curing Behaviour in Resin Injection Pultrusion. *Proceedings: ECCM18*, 2018.